WEST

□ Generate Collection Print

L14: Entry 9 of 14 File: USPT

US-PAT-NO: 5455943
DOCUMENT-IDENTIFIER: US 5455943 A

TITLE: Method and apparatus for finding longest and closest matching string in history buffer prior to current string

DATE-ISSUED: October 3, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chambers, IV; Lloyd L.	Menlo Park	CA		

US-CL-CURRENT: 707/100; 341/51, 708/203

CLAIMS:

What is claimed is:

1. A data processing apparatus for compressing source data, the apparatus comprising:

(a) memory means for storing data in identifiable locations thereof, said stored data including at least part of the source data that is to be compressed; and

(b) a processing unit operatively coupled to the memory means for processing data within the memory means, the processing unit including current-specifying means for specifying as current, a particular location within the memory means that stores a corresponding portion of the source data;

(a.1). wherein the memory means includes a history portion for storing one or more data strings that define historical source data;

(a.2) wherein the memory means further includes a fast-path portion for storing groups of one or more history pointers, where each group is dedicated to a respective one of a prespecified plurality of unique code sequences and the one or more history pointers of that group respectively point to and identify a corresponding one or more locations within the memory means that store one or more of the historical data strings, if any, which contain the respective one of the prespecified plurality of unique code sequences to which the corresponding group is dedicated and, in cases where more than one of the identified historical data strings has the same prespecified sequence of code, the history pointers of the corresponding group are arranged to indicate the positional order of the more than one identified historical data strings relative to the current location so that it can be quickly determined from said arrangement of the history pointers whether a first history location that stores a corresponding first of the identified historical data strings is address-wise closer to or further from the current location than is a second history location that stores a corresponding second of the identified historical data strings.

2. A data processing apparatus according to claim 1 wherein:

each of said plural groups is defined by a respective list stored in the memory means and each list contains one or more records;.

each record includes a corresponding one of said history pointers; and

in a case where a given list contains more than one record, the plural records of the given list are ordered such that their order indicates the positional order of the corresponding more than one pointed-to memory locations relative to the current location.

3. A data processing apparatus according to claim 2 wherein each of the plural lists is a linked list and each record of each linked list includes a next-record pointer for either pointing to a next record within the linked list or indicating the end of the list in the case where there is no next record.

4. A data processing apparatus according to claim 1 wherein the identifiable locations in the history portion of said memory means each store a dataword of prespecified bit length and the prespecified sequence of code includes two or more adjacent datawords.

5. A data processing apparatus according to claim 1

wherein the memory means further stores program instructions which direct the processing unit to search said history portion of the the memory means to find, with respect to a current string of data starting at the current location, first a set of one or more data strings within the history portion that match the starting and subsequent successive parts of the current String to the longest extent possible, and if more than one such longest matching data string is found, identifying the longest matching data string that is also stored in a location that is address-wise closest to the starting location of the current string; and

wherein the processing unit is responsive to said program instructions.

6. A data processing apparatus according to claim 5 wherein said direct the processing unit to generate a vector representing an offset from the current location to the location of the longest and closest matching data string within the history portion.

7. A data processing apparatus according to claim 6 wherein the generated vector is a compression vector for replacing the matched starting and subsequent successive parts of the current string and the generated compression vector consists of fewer bits than that portion of the current string which it replaces.

8. A data processing apparatus according to claim 1 wherein each of the historical data strings is limited to a maximum length of 20 bytes.

9. A data processing apparatus according to claim 1

wherein the processing unit generates, for a current string in the source data having a length of no more than 20 bytes, a corresponding compression vector having a length of no more than 5 bits in the case where a matching historical data string is found in said history portion of the memory means; and

wherein the processing unit finds the matching historical data string, if any, by using one of said groups of one or more history pointers.

10. A data processing apparatus according to claim 1

wherein the processing unit generates, for a current string in the source data having a predefined code sequence at a prespecified portion thereof, a corresponding, but smaller sized, compression vector representing at least the predefined code sequence in the case where a matching historical data string is found in said history portion of the .memory means; and

wherein the processing unit tries to find the matching historical data string, if any, by using said groups of one or more dedicated history pointers.

11. A data processing apparatus according to claim 10

wherein the processing unit determines if there is at least one historical data string in the history portion having the predefined code sequence at a prespecified portion thereof, such that the at least one historical data string matches at least partly with the current string, by testing the fast-path portion to determine if there exists among said dedicated groups of pointers, a group dedicated to the predefined code sequence at the prespecified portion of the current string.

12. A data processing apparatus according to claim 11

wherein, for the case where the processing unit determines that there is at least one historical data string in the history portion having the predefined code sequence because a corresponding dedicated group of pointers is found in the fast-path portion, the processing unit next determines if the matching one or more historical data strings, which are pointed to by the corresponding group of one or more pointers, each match the current string to a greater extent without comparing the predefined code sequence at the prespecified portion of the current string against the predefined code sequence at the prespecified portion of each of the matching one or more historical data strings, given that the match between said prespecified portions of the current string and the matching one or more historical data strings is already established by the existence of the corresponding dedicated group of pointers.

13. A data processing apparatus according to claim 12

wherein the respective prespecified portions of the current string and the matching one or more historical data strings are the start sequences of said strings.

14. A data processing apparatus according to claim 13

wherein each respective pointer in said groups of one or more dedicated history pointers points to the location of the start sequence of the corresponding historical data string.

15. A data processing apparatus according to claim 11

wherein the fast-path portion of the memory means includes a data look-up table that indicates for every code permutation possible at the prespecified portion of the current string, a corresponding indication of whether or not a respective, dedicated group of pointers exists in the fast-path portion; and

wherein the processing unit uses the data look-up table to determine if there exists among said dedicated groups of pointers, a group dedicated to the predefined code sequence.

16. A data processing apparatus according to claim 11

wherein the fast-path portion of the memory means includes a data look-up table that indicates for every code permutation possible at the prespecified portion of the current string, a corresponding pointer to the location, if any, of the respective, dedicated group of pointers; and wherein the processing unit uses the data look-up table to locate the group dedicated to the predefined code sequence.

17. A machine-implemented data processing method comprising the steps of:

(a) storing a plurality of historical data strings within memory;

(b) generating within memory, plural groups of one or more pointers, where each group is dedicated to a respective one of a prespecified plurality of unique code sequences and the one or more pointers of that group respectively point to and identify a corresponding one or more locations within memory that store respective ones of the historical data strings that contain a corresponding code sequence to which that group is dedicated; and

(c) ordering the pointers of each dedicated group that has plural pointers to indicate the positional order in memory of the corresponding historical data strings so that, in cases where more than one of the historical data strings has the same prespecified sequence of code, the ordering of the corresponding pointers indicates the relative positional order of the respective more than one historical data strings relative to one another within the memory.

18. A machine-implemented data processing method according to claim 17 further comprising the steps of:

(d) identifying a position within said memory as a current position; and

(e) creating a fast-path array in said memory where the fast-path array includes one or more lists each associated with a unique prespecified code sequence; where each list contains a head record pointing to a first of the plural historical data strings that contains the associated unique code sequence and where, for the case wherein a given list points to more than one of the plural historical data strings, said first historical data string is the one for the given list which is stored address-wise closest to the current position.

19. A machine-implemented data processing method according to claim 18 further comprising the steps of:

(f) storing at the current position a current data string having at least first and second code subsequences;

(g) combining the first and second code subsequences to define an index-pair code;

(h) converting the index-pair code into a head-record pointer that points to a corresponding head record within the fast-path array; and

(i) using the corresponding head record to locate among said historical data strings, one or more matching historical data strings, if any, that each contains the associated unique code sequence, and if there are more than one matching historical data strings, using the ordering of the pointers to identify the longest matching one of the historical data strings which is closest to the current position.

20. A machine-implemented data processing method according to claim 19 further comprising the step of:

generating a compression vector representing an offset from the current location to the located instance of the longest matching and closest historical data string.

21. A data processing system comprising:

(a) first storing means for storing in addressable first storage locations of a memory, a plurality of historical data strings, each historical data string having, at a prespecified portion thereof, a predefined number of successive bits defining a corresponding code sequence;

(b) pointer generating means for generating a plurality of pointers each pointing to a storage location of a respective one of said historical data strings;

(c) pointer organizing means for organizing said pointers into dedicated groups,

wherein each dedicated group is dedicated to a respective one of a prespecified plurality of unique code sequences that can define said corresponding code sequences of the historical data strings, and

wherein each dedicated group consists of one or more of the generated pointers that point to a respective one or more of the historical data strings that each contains the corresponding unique code sequence at the prespecified portion thereof; and

(d) pointer ordering means for ordering the pointers of dedicated groups having more than one pointer such that the ordering of pointers with each such dedicated group indicates the address-wise storage order of the corresponding more than one pointed-to historical data strings relative to one another.

22. A data processing system according to claim 21, further comprising:

(e) second storing means for storing in addressable second storage locations of said memory, one or more new data strings;

(f) current designating means for designating as current, a first of the one or more new data string;

(g) matching means for locating within a predefined window portion of said first storage locations, one or more longest matching ones of the historical data strings that each match the current data string to a maximal extent, said matching means using the one pointer or ordered plural pointers of the respective dedicated group to identify the one or more longest matching historical data strings.

23. A data processing system according to claim 22, wherein, for the case where there is more than one longest matching historical data string, the matching means uses the ordering of the pointers in the corresponding dedicated group to identify a first of the plural longest matching historical data strings which is address-wise closest to the current string.

24. A data processing system according to claim 23, further comprising:

(h) compression vector generating means for generating, in response to the identification by the matching means of the address-wise closest and longest matching historical data string within the predefined window portion, a compression vector having an offset portion representing an address offset between the storage location of the current string and the storage location of the identified longest matching and closest historical data string and further having a length portion representing the corresponding length of match.

25. A data processing system according to claim 21, further comprising:

(e) indexing means for converting a supplied code sequence into a group pointer that points to the dedicated group corresponding to the supplied code sequence.

26. A data processing system according to claim 25, further comprising:

(f) second storing means for storing in addressable second storage locations of said memory, one or more new data strings;

(g) current designating means for designating as current, a first of the one or more new data string;

(h) matching means for locating within a predefined window portion of said first storage locations, one or more longest matching ones of the historical data strings that each match the current data string to a maximal extent, said matching means using the indexing means to locate the respective data dedicated group and using the one pointer or ordered plural pointers of the respective dedicated group to identify the one or more longest matching historical data strings.

27. A data processing system according to claim 26, wherein, for the case where there is more than one longest matching historical data string, the matching means uses the ordering of the pointers in the corresponding dedicated group to identify a first of the plural longest matching historical data strings which is address-wise closest to the current string.

28. A data processing system according to claim 27, further comprising:

(i) compression vector generating means for generating, in response to the identification by the matching means of the address-wise closest and longest matching historical data string within the predefined window portion, a compression vector having an offset portion representing an address offset between the storage location of the current string and the storage location of the identified longest matching and closest historical data string and further having a length portion representing the corresponding length of match.